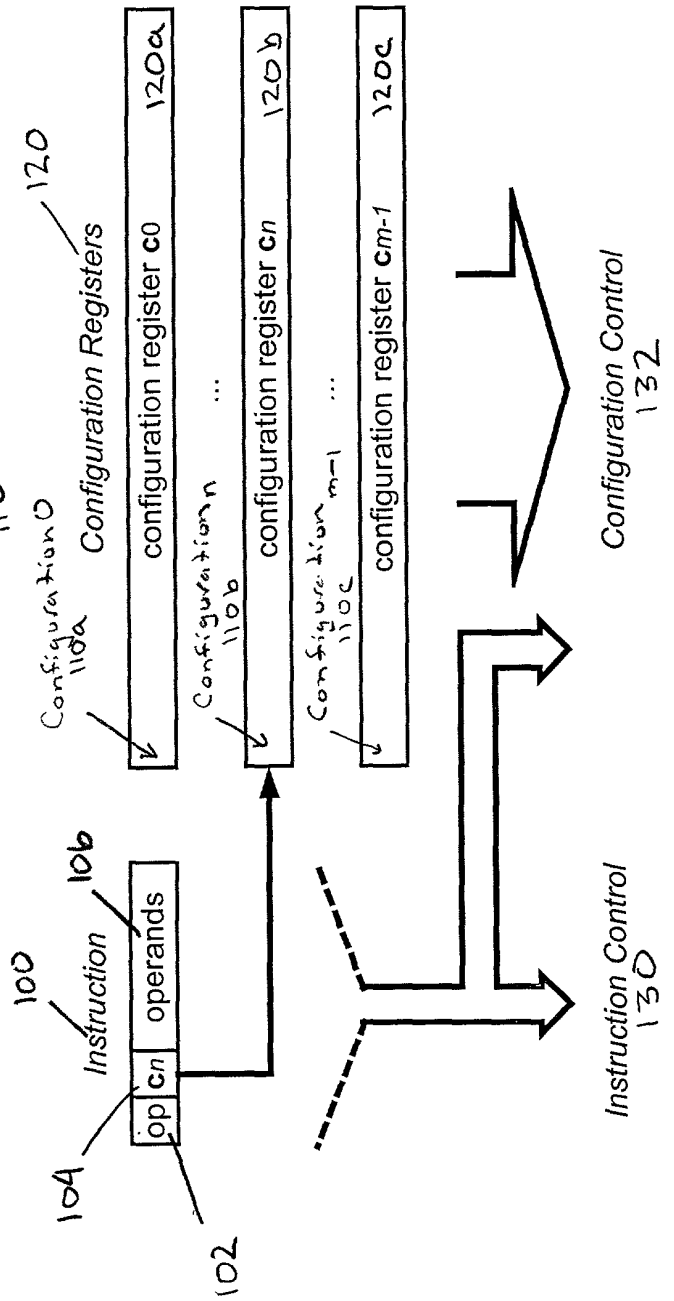
100
Instruction
106
op
cn
operands
104
102
Configuration Registers
120
Configuration 0
1108
Configuration n
1109
Configuration m-1
110c
configuration register c0
120a
configuration register cn
120b
configuration register cm-1
120c
Instruction Control
130
Configuration Control
132

Fig. 1

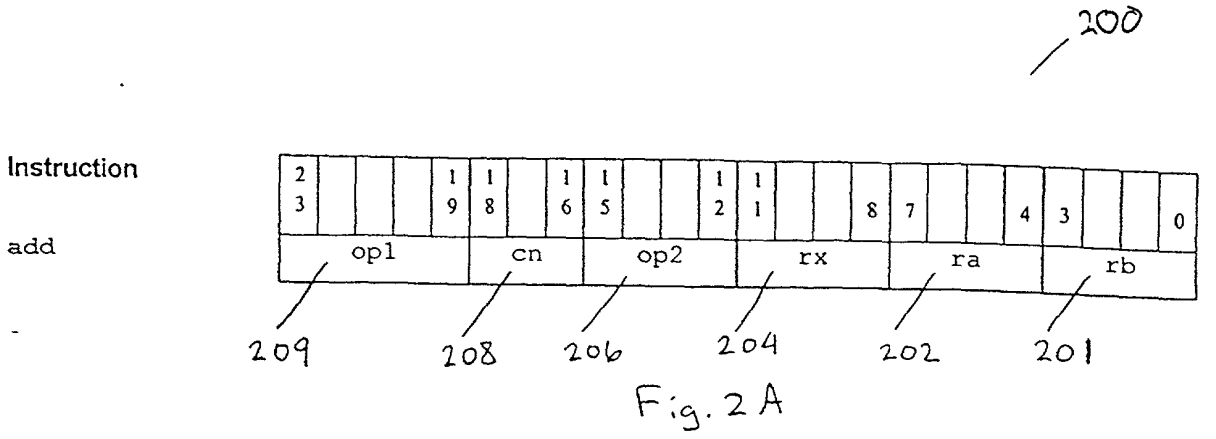
200
Instruction
add
23 19 18 16 15 12 11 8 7 4 3 0
op1
cn
op2
rx
ra
rb
209
208
206
204
202
201

Fig. 2A

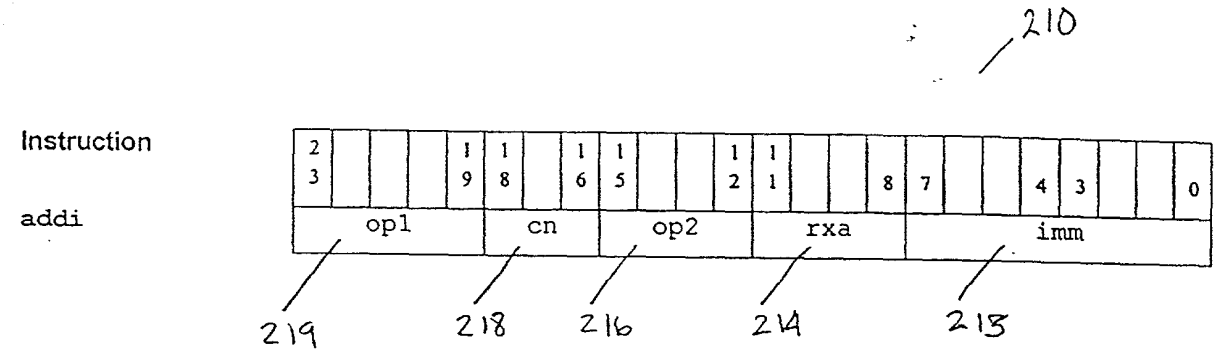
210
Instruction
addi
23 19 18 16 15 12 11 8 7 4 3 0
op1
cn
op2
rxa
imm
219
218
216
214
213

Fig. 2B

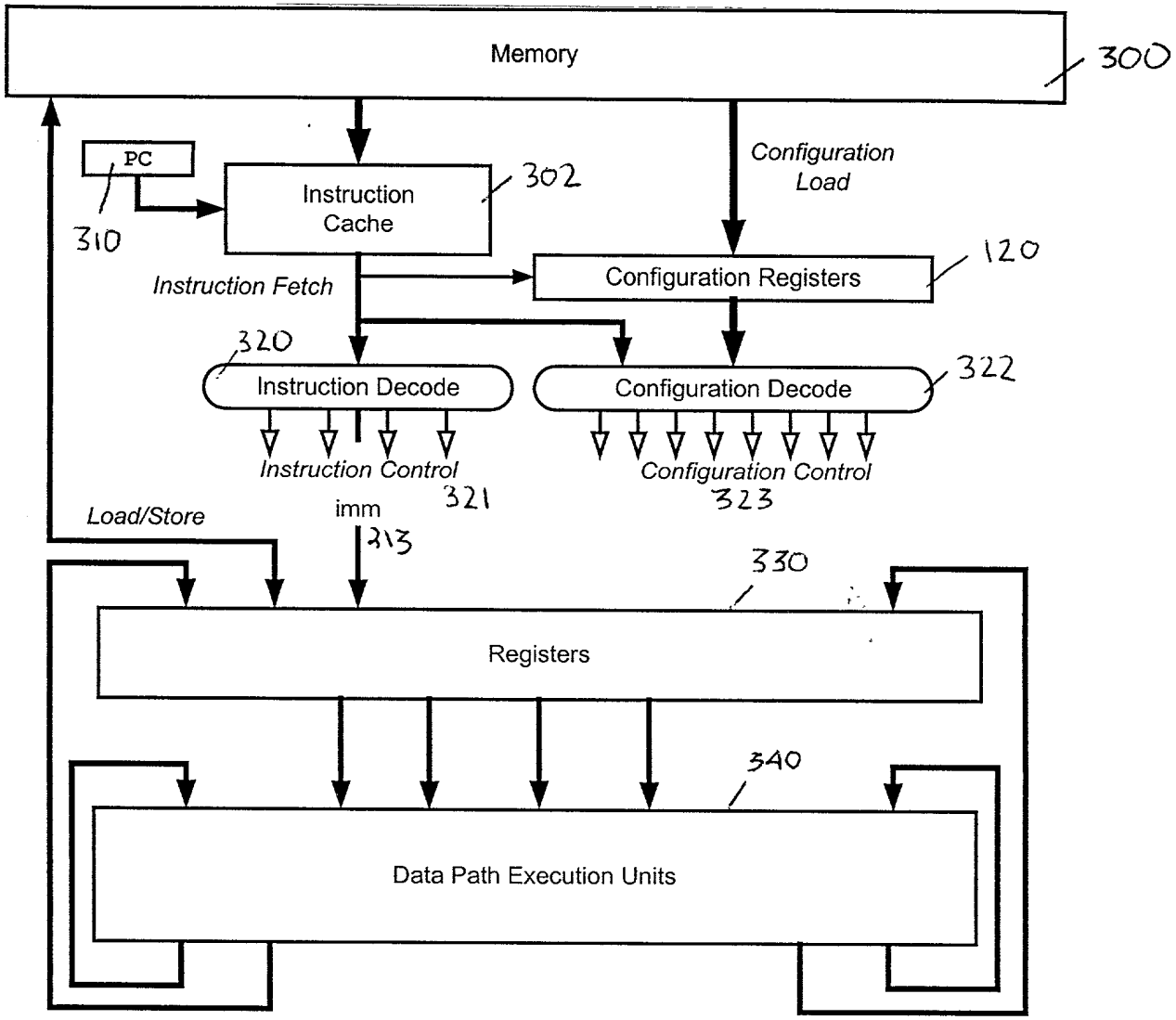
Memory
300
PC
310
Instruction Cache
302
Configuration Load
Instruction Fetch
Configuration Registers
120
320
Instruction Decode
Configuration Decode
322
Instruction Control
321
Configuration Control
323
Load/Store
imm
313
330
Registers
340
Data Path Execution Units

Fig. 3

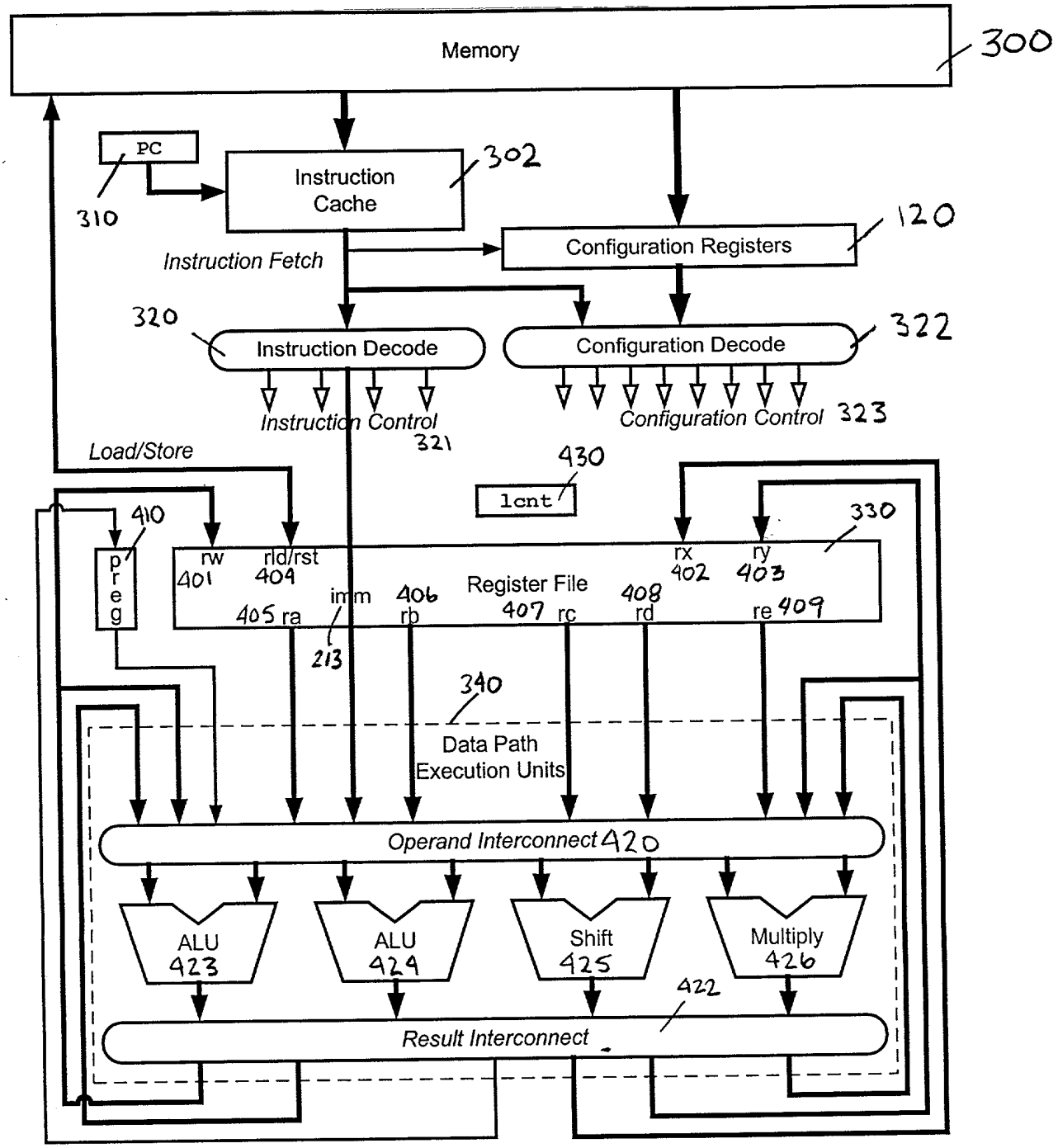
Memory
300
PC
310
Instruction Cache
302
Configuration Registers
120
Instruction Fetch
320
Instruction Decode
Configuration Decode
322
Instruction Control
321
Configuration Control
323
Load/Store
430
lcnt
410
330
preg
rw
401
rld/rst
404
rx
402
ry
403
Register File
407
405 ra
imm
213
406
rb
408
rd
re 409
rc
340
Data Path
Execution Units
Operand Interconnect 420
ALU
423
ALU
424
Shift
425
Multiply
426
422
Result Interconnect

Fig. 4

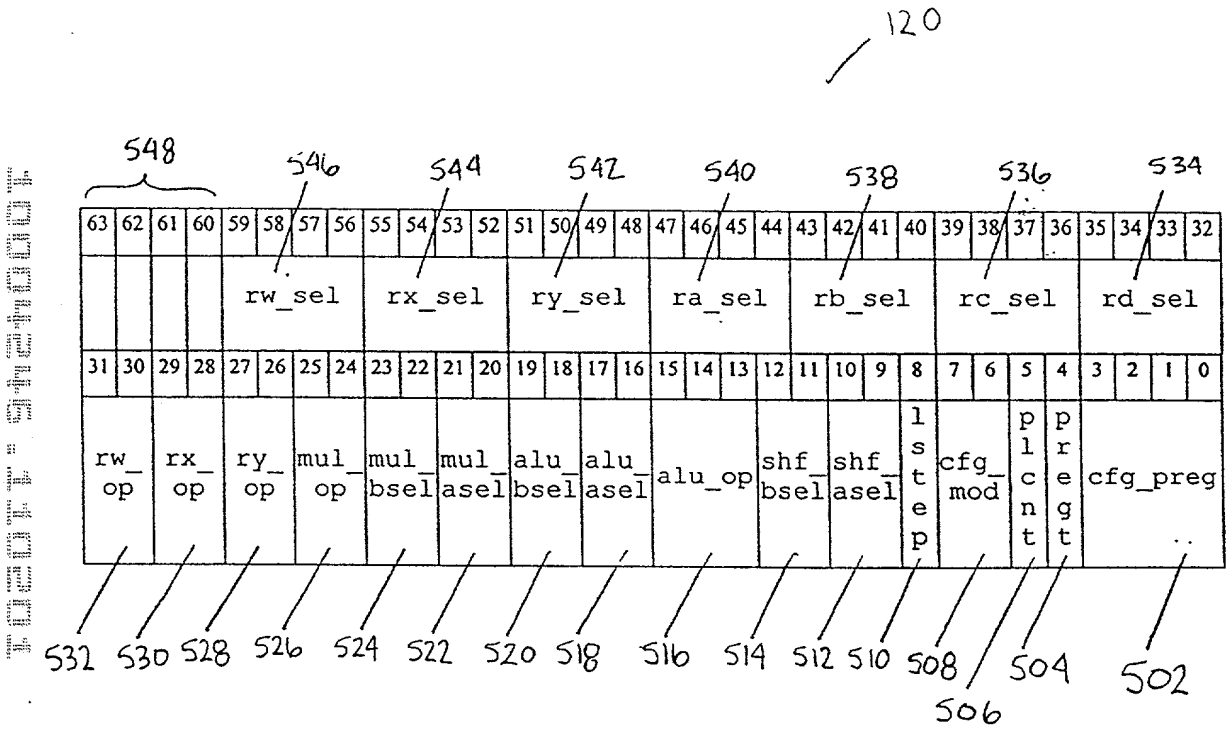
120
548
546
544
542
540
538
536
534
63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32
rw_sel
rx_sel
ry_sel
ra_sel
rb_sel
rc_sel
rd_sel
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
rw_op
rx_op
ry_op
mul_op
mul_bsel
mul_asel
alu_bsel
alu_asel
alu_op
shf_bsel
shf_asel
step
cfg_mod
plcnt
pregt
cfg_preg
532 530 528 526 524 522 520 518 516 514 512 510 508 506 504 502

Fig. 5A

	Field	Bits	Description
546	rw_sel	59:56	Register W port selection, r0 – r15.
544	rx_sel	55:52	Register X port selection, r0 – r15.
542	ry_sel	51:48	Register Y port selection, r0 – r15.
540	ra_sel	47:44	Register A port selection, r0 – r15.
538	rb_sel	43:40	Register B port selection, r0 – r15.
536	rc_sel	39:36	Register C port selection, r0 – r15.
534	rd_sel	35:32	Register D port selection, r0 – r15.
532	rw_op	31:30	Register W write select. 0 = no write, 1 = alu_out, 2 = shift_out, 3 = mul_out.
530	rx_op	29:28	Register X write select. 0 = no write, 1 = alu_out, 2 = shift_out, 3 = mul_out.
528	ry_op	27:26	Register Y write select. 0 = no write, 1 = alu_out, 2 = shift_out, 3 = mul_out.
526	mul_op	25:24	Multiply high/low word select. 0 = lo *lo, 1 = lo* hi, 2 = hi * lo, 3 = hi * hi.
524	mul_bsel	23:22	Multiplier operand B select. 0 = ra, 1 = rb, 2 = rc, 3 = rd.
522	mul_asel	21:20	Multiplier operand A select. 0 = ra, 1 = rb, 2 = rc, 3 = rd.
520	alu_bsel	19:18	ALU operand B select. 0 = ra, 1 = rb, 2 = rc, 3 = rd.
518	alu_asel	17:16	ALU operand A select. 0 = ra, 1 = rb, 2 = rc, 3 = rd.
516	alu_op	15:13	ALU operation: 0 = pass, 1 = add, 2 = sub, 3 = min, 4 = max, 5 = and, 6 = or, 7 = xor.
514	shf_bsel	12:11	Shift operand B select. 0 = ra, 1 = rb, 2 = rc, 3 = rd.
512	shf_asel	10:9	Shift operand A select. 0 = ra, 1 = rb, 2 = rc, 3 = rd.
510	lstep	8 :8	Step (decrement) loop counter lcnt.
508	cfg_mod	7:6	Configuration modifier select. 0 = ALU op mod, 1 = rw op mod, 2 = rw mod, 3 = rc mod.
506	plcnt	5:5	Predicate execution of configuration on non-zero loop count lcnt != 0.
504	pregt	4:4	Specifies the 1-bit value in predicate register cfg_preg that enables execution of this configuration.
502	cfg_preg	3:0	Predicate execution of configuration on value of predicate register `cfg_preg == pregt` value.

Fig. 5B

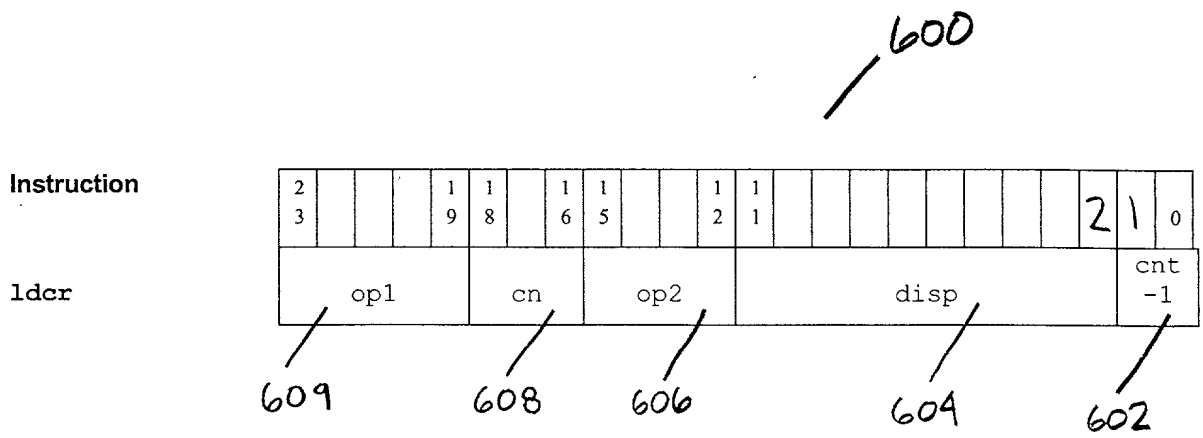
600
Instruction
23
19
18
16
15
12
11
2
1
0
ldcr
op1
cn
op2
disp
cnt
-1
609
608
606
604
602

Fig. 6

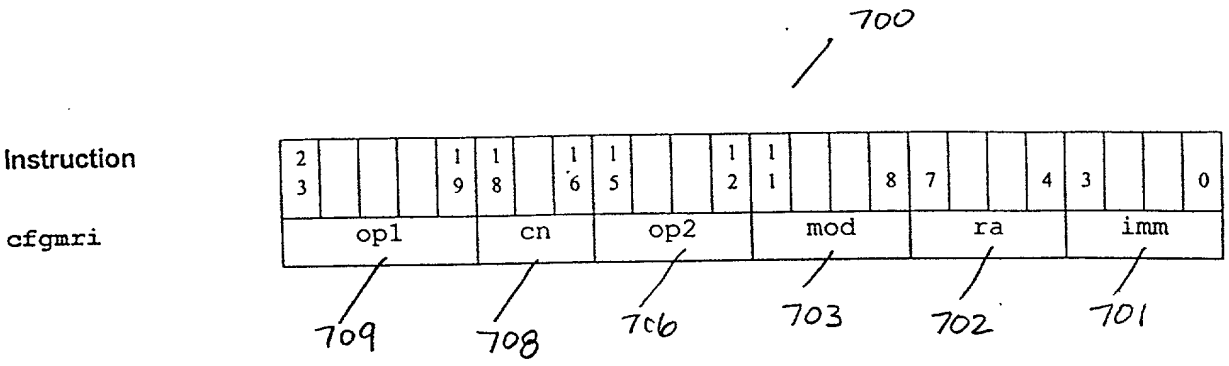
700
Instruction
cfgmri
23
19
18
16
15
12
11
8
7
4
3
0
op1
cn
op2
mod
ra
imm
709
708
706
703
702
701

Fig. 7A

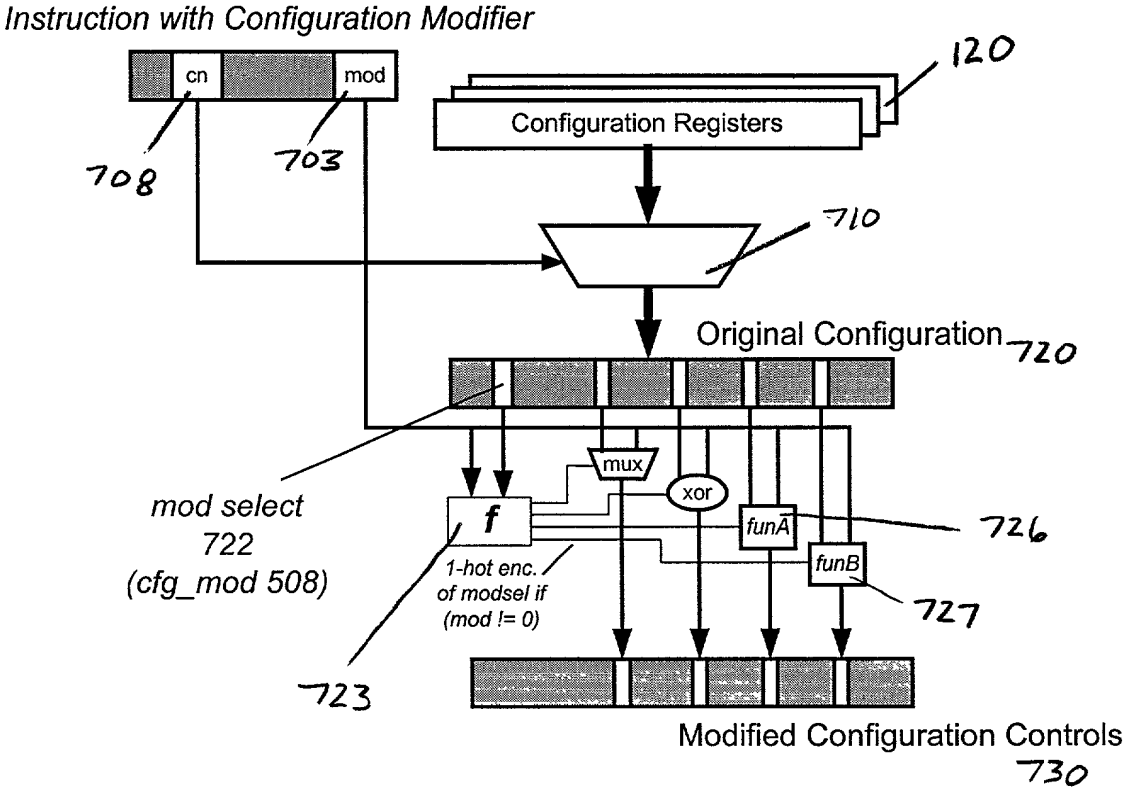
Instruction with Configuration Modifier
cn
mod
708
703
120
Configuration Registers
710
Original Configuration
720
mux
xor
f
funA
funB
726
727
mod select
722
(cfg_mod 508)
1-hot enc.
of modsel if
(mod != 0)
723
Modified Configuration Controls
730

Fig. 7B

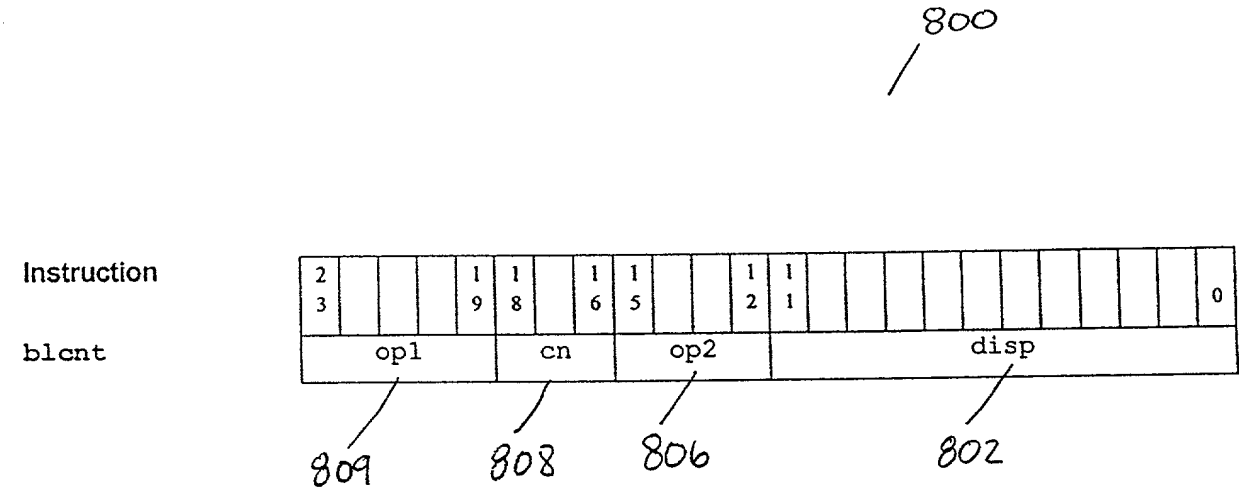
800
Instruction
blcnt
23
19
18
16
15
12
11
0
op1
cn
op2
disp
809
808
806
802

Fig. 8

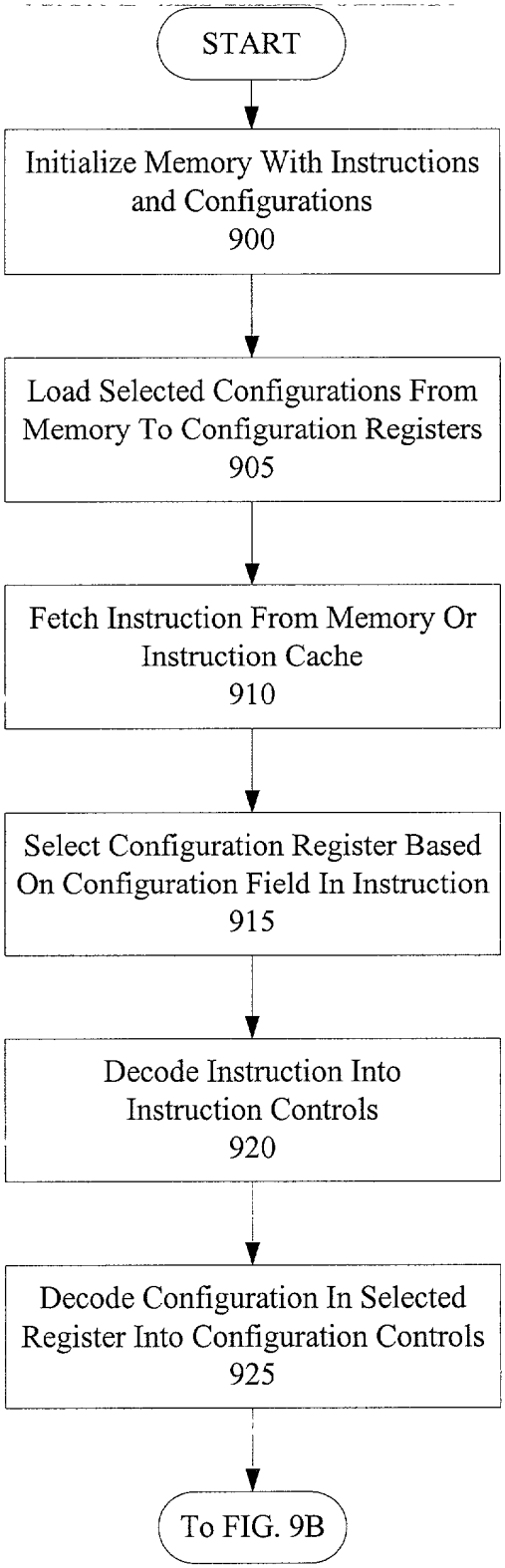
START
Initialize Memory With Instructions and Configurations
900
Load Selected Configurations From Memory To Configuration Registers
905
Fetch Instruction From Memory Or Instruction Cache
910
Select Configuration Register Based On Configuration Field In Instruction
915
Decode Instruction Into Instruction Controls
920
Decode Configuration In Selected Register Into Configuration Controls
925
To FIG. 9B

FIG. 9A

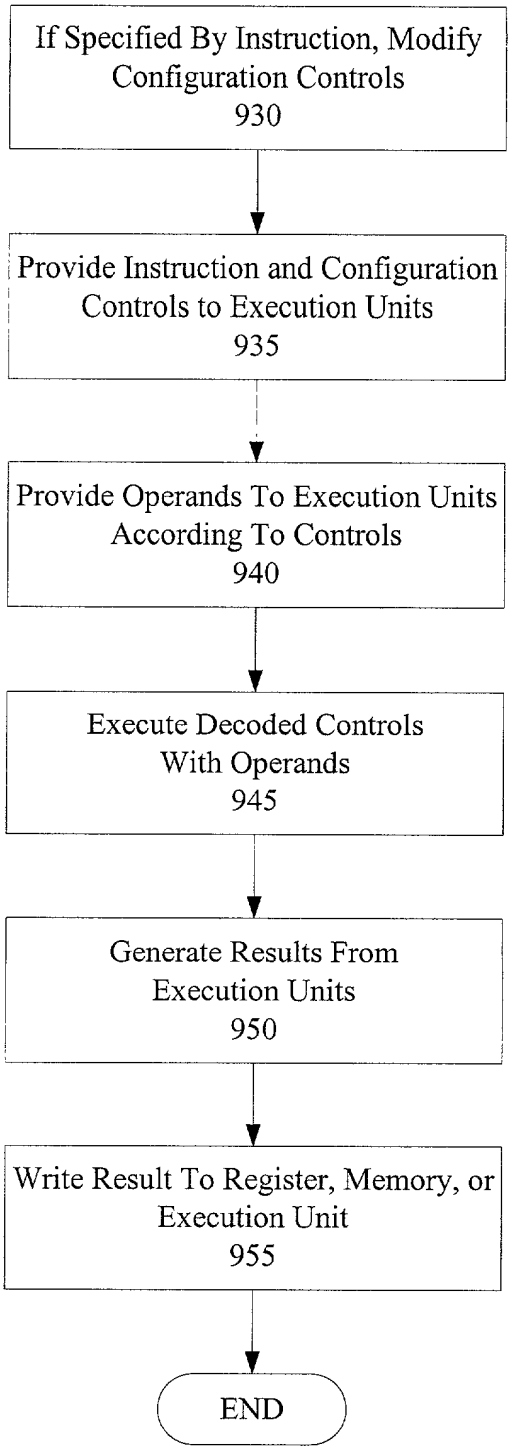
If Specified By Instruction, Modify Configuration Controls
930
Provide Instruction and Configuration Controls to Execution Units
935
Provide Operands To Execution Units According To Controls
940
Execute Decoded Controls With Operands
945
Generate Results From Execution Units
950
Write Result To Register, Memory, or Execution Unit
955
END

FIG. 9B

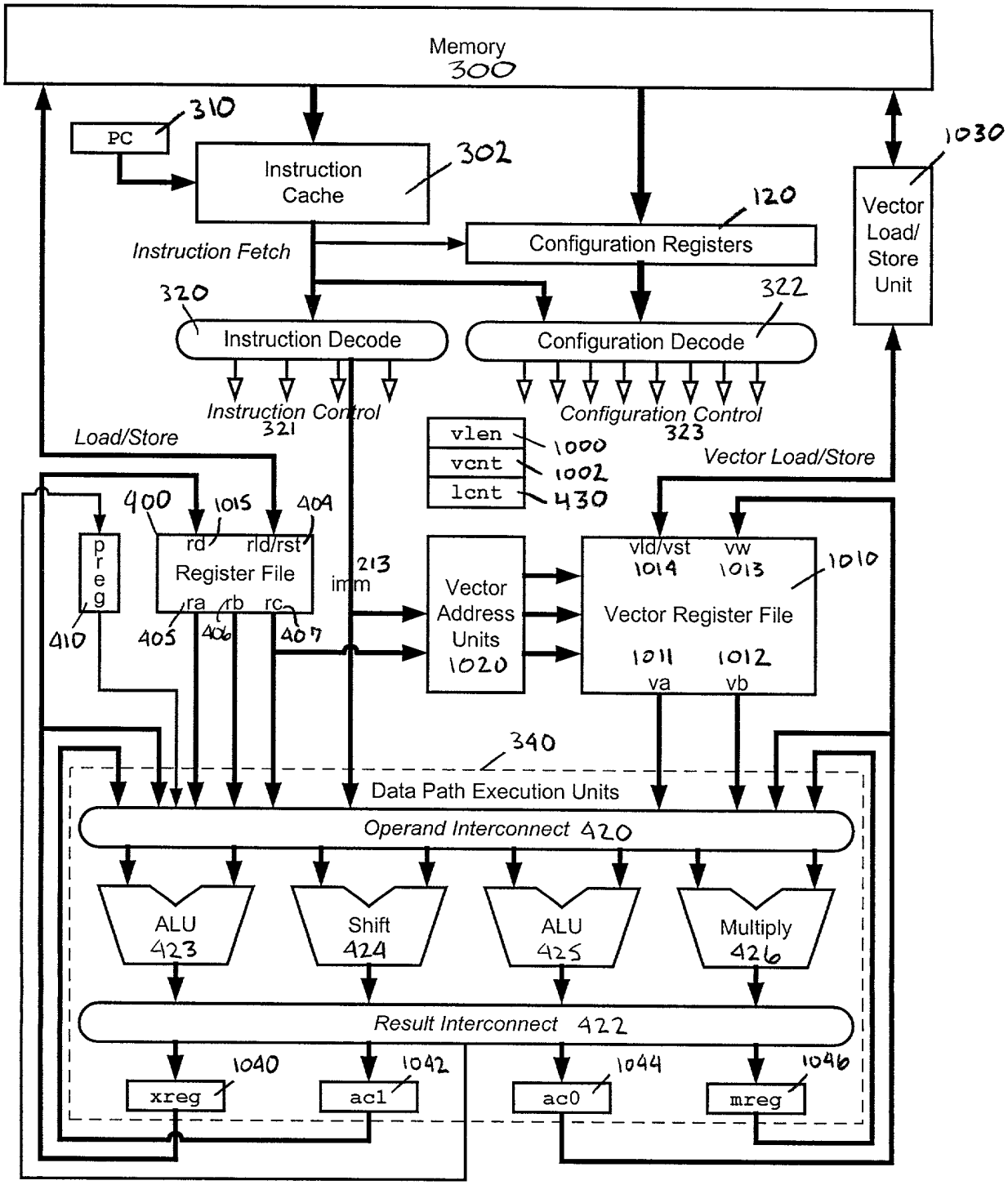
Memory
300
PC
310
Instruction Cache
302
Instruction Fetch
Configuration Registers
120
Vector Load/ Store Unit
1030
320
Instruction Decode
322
Configuration Decode
Instruction Control
321
Configuration Control
323
Load/Store
Vector Load/Store
vlen
vcnt
lcnt
1000
1002
430
400
1015
404
p r e g
rd
rld/rst
Register File
ra
rb
rc
410
405
406
407
213
imm
Vector Address Units
1020
vld/vst
1014
vw
1013
1010
Vector Register File
1011
va
1012
vb
340
Data Path Execution Units
Operand Interconnect 420
ALU
423
Shift
424
ALU
425
Multiply
426
Result Interconnect 422
1040
1042
1044
1046
xreg
ac1
ac0
mreg

Fig. 10

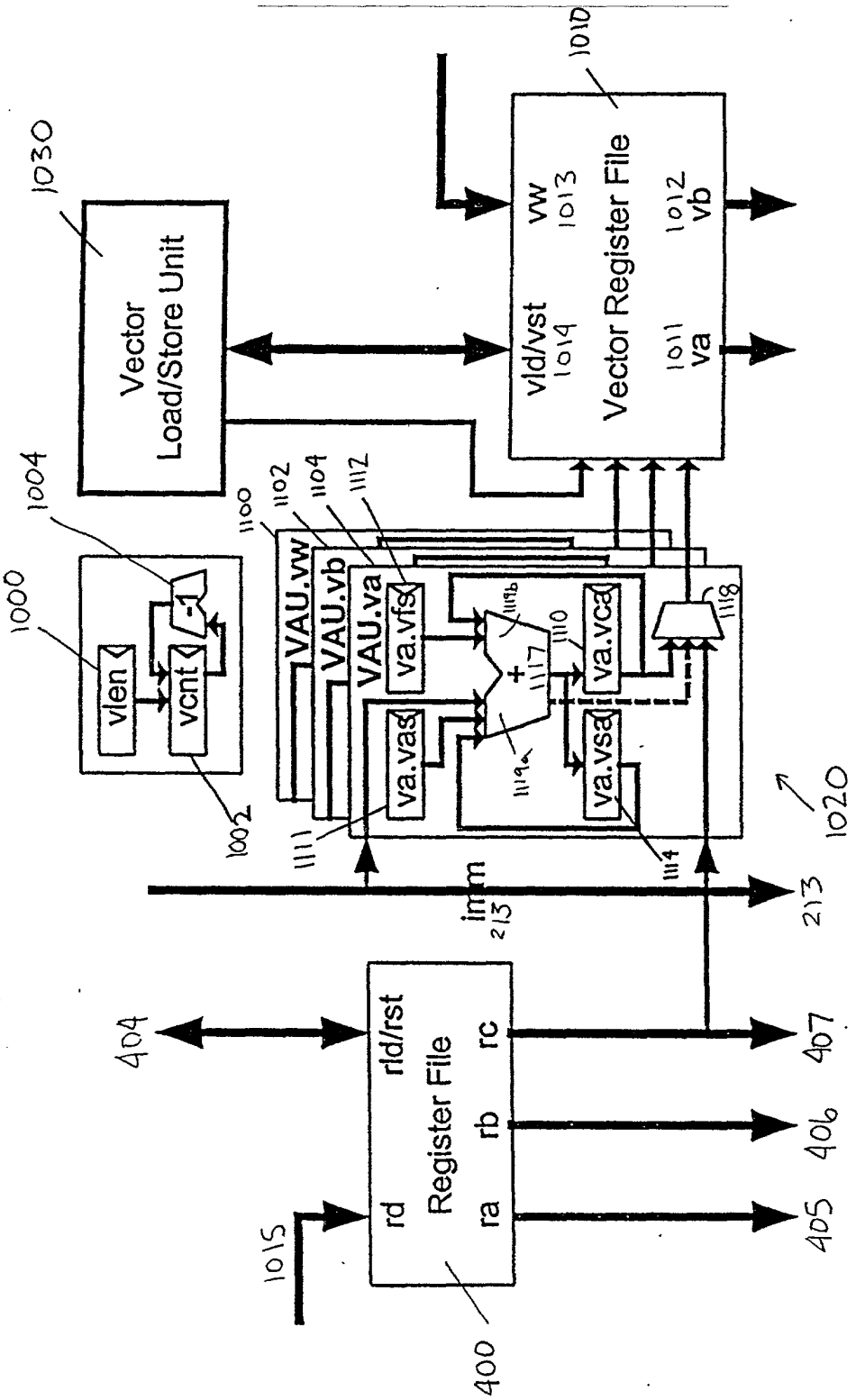
1010
Vector Register File
vw
1013
1012
vb
vld/vst
1014
1011
va
1030
Vector
Load/Store Unit
1004
1000
vlen
vcnt
-1
1002
1100
1102
1104
1112
VAU.vw
VAU.vb
VAU.va
va.vfs
va.vas
va.vca
va.vsa
1116b
1110
1117
1118
1116a
1111
1114
1020
imm
213
213
404
rld/rst
Register File
rd
ra
rc
rb
407
406
405
1015
400

Fig. 11

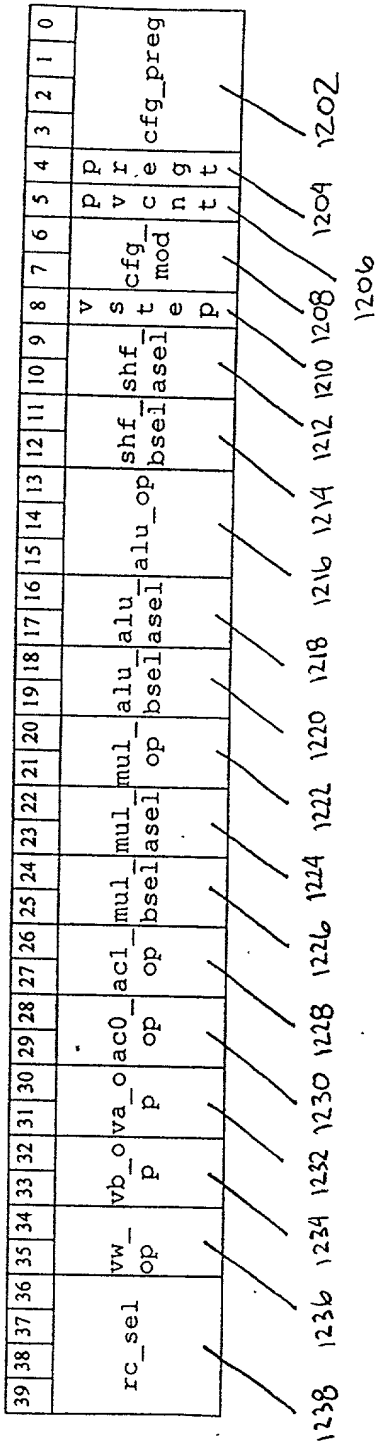
1200
1202
1204
1206
1208
1210
1212
1214
1216
1218
1220
1222
1224
1226
1228
1230
1232
1234
1236
1238
cfg_preg
ppvcngtt
cfg_mod
vstep
shf_asel
shf_bsel
alu_op
alu_asel
alu_bsel
mul_op
mul_asel
mul_bsel
acl_op
ac0_op
va_op
vb_op
vw_op
rc_sel
39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Fig. 12A

	Field	Bits	Description
1238	rc_sel	39:36	Register C port selection, r0 – r15.
1236	vw_op	35:34	Vector register W port operation. 0 = no write, hold address vw.vca; 1 = write address rc, hold address vw.vca; 2 = write word, hold address vw.vca; 3 = write word, step current address vw.vca.
1234	vb_op	33:32	Vector register B port operation. 0 = no read, hold address vb.vca; 1 = read address rc, hold current address vb.vca; 2 = read word, hold address vb.vca; 3 = read word, step current address vb.vca.
1232	va_op	31:30	Vector register A port operation. 0 = no read, hold address va.vca; 1 = read address rc, hold current address va.vca; 2 = read word, hold address va.vca; 3 = read word, step current address va.vca.
1230	ac0_op	29:28	Accumulator 0 operation. 0 = hold; 1 = write ALU output, 2 = write Multiplier output, 3 = write shift output.
1228	ac1_op	27:26	Accumulator 1 operation. 0 = hold; 1 = write ALU output, 2 = write Multiplier output, 3 = write shift output.
1226	mul_bsel	25:24	Multiplier operand B select. 0 = vb, 1 = ac0, 2 = rb, 3 = rc.
1224	mul_asel	23:22	Multiplier operand A select. 0 = va, 1 = ac0, 2 = ra, 3 = rc.
1222	mul_op	21:20	Multiplier operation. 0 = A, B signed; 1 = A signed, B unsigned, 2 = A unsigned, B signed, 3 = A, B unsigned.
1220	alu_bsel	19:18	ALU operand B select. 0 = vb, 1 = ac0, 2 = rb, 3 = ac1.
1218	alu_asel	17:16	ALU operand A select. 0 = va, 1 = ac0, 2 = ra, 3 = mreg.
1216	alu_op	15:13	ALU operation: 0 = pass, 1 = add, 2 = sub, 3 = min, 4 = max, 5 = and, 6 = or, 7 = xor.
1214	shf_bsel	12:11	Shift operand B select. 0 = vb, 1 = ac0, 2 = rb, 3 = rc.
1212	shf_asel	10:9	Shift operand A select. 0 = va, 1 = ac0, 2 = ra, 3 = ac1.
1210	vstep	8:8	Step (decrement) the vector counter vcnt.
1208	cfg_mod	7:6	Configuration modifier select. 0 = ALU op mod, 1 = VAU op mod, 2 = rc mod, 3 = ac op mod.
1206	pvcnt	5:5	Predicate execution of configuration on non-zero vector count vcnt != 0.
1204	pregt	4:4	Specifies the 1-bit value in predicate register cfg_preg that enables execution of this configuration.
1202	cfg_preg	3:0	Predicate execution of configuration on value of predicate register cfg_preg ==. pregt value.

Fig. 12B

Attorney: Craig A. Gelfound
Docket No.: 42113/CAG/B600
Inventor(s): John R. Nickolls, et al
Title: RECONFIGURABLE PROCESSING SYSTEM AND METHOD
Sheet 16 of 17

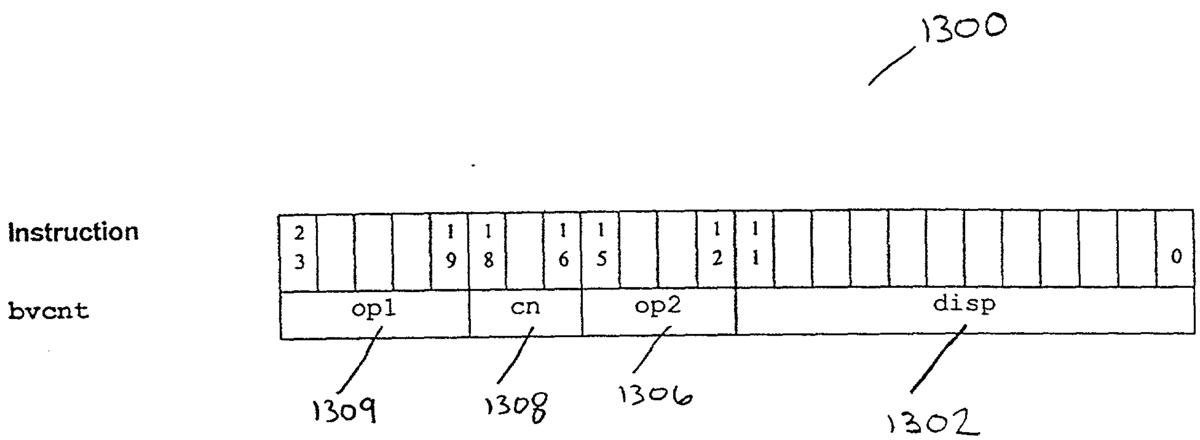
1300
Instruction
bvcnt
23
19
18
16
15
12
11
0
op1
cn
op2
disp
1309
1308
1306
1302

Fig. 13